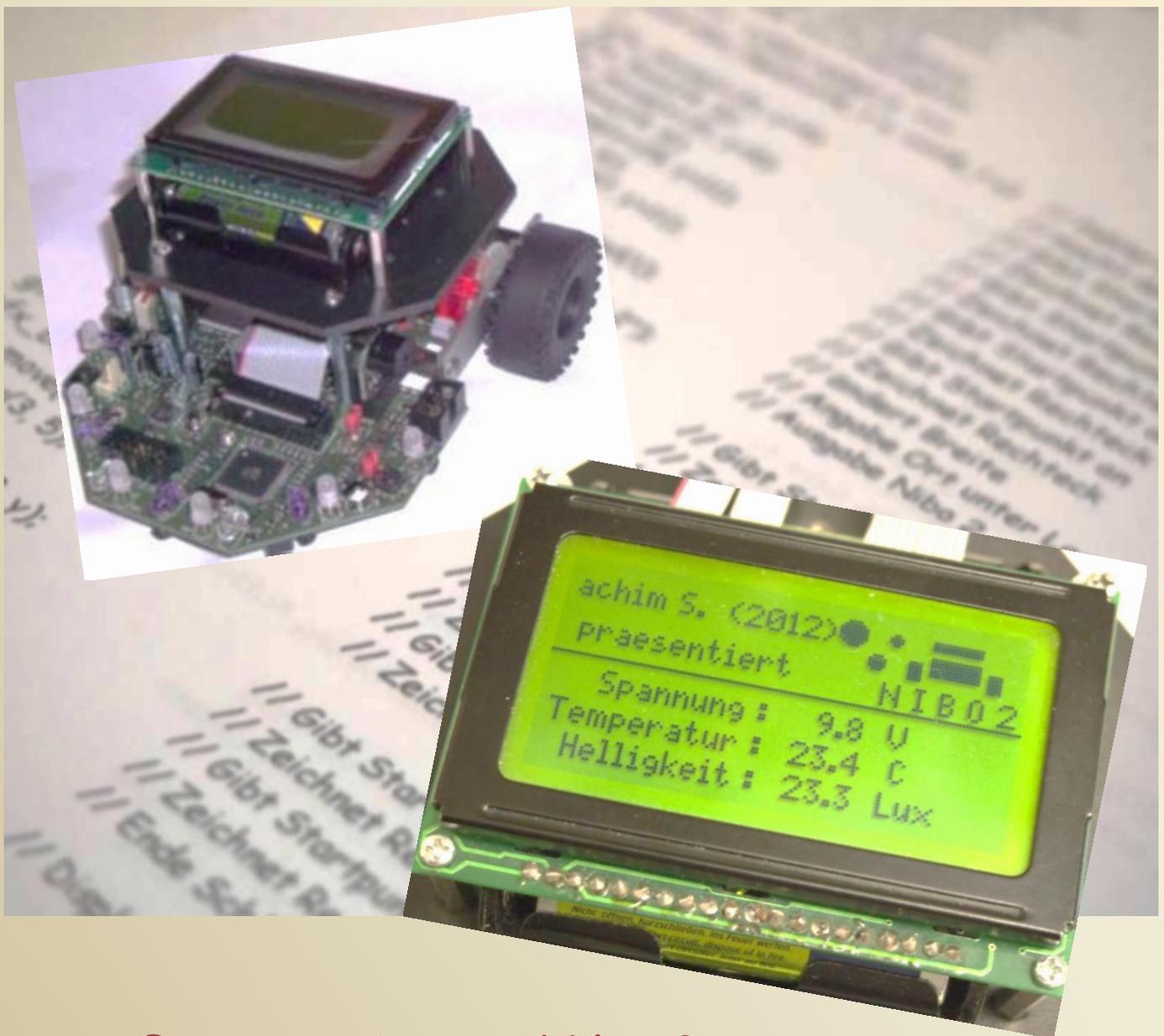


NIBO 2*

© by HJS

PROGRAMMIERUNG



Programmierung Nibo 2
Teil 1 - ATmel AVR Studio 6

* by nicai-
systems

Nibo 2

Programmierung Teil 1 - ATmel AVR Studio 6

Ich arbeite mit Windows 7 und den angegebenen Programmen.
Es können auch andere Programme, z.B. Linux, verwendet werden.
Welches Programm ihr nutzt, müsst ihr selber entscheiden.
Die Beispielprogramme findet ihr bei Roboter.cc. Alle Programme habe ich auf Funktion getestet. Alle Fotos bei hjs. (Stand 31012013)

1. Atmel AVR Studio 6 - Neues Projekt
2. Atmel AVR Studio 6 - Kurzanleitung - Neues Projekt
3. Atmel AVR Studio 6 - Import eines Projekt von AVR Studio 4
4. Atmel AVR Studio 6 - Import eines Projektes als Text
5. Atmel AVR Studio 6 - Software und mögliche Fehler
6. Atmel AVR Studio 6 - Windows 8



by **H.J. Seeger**

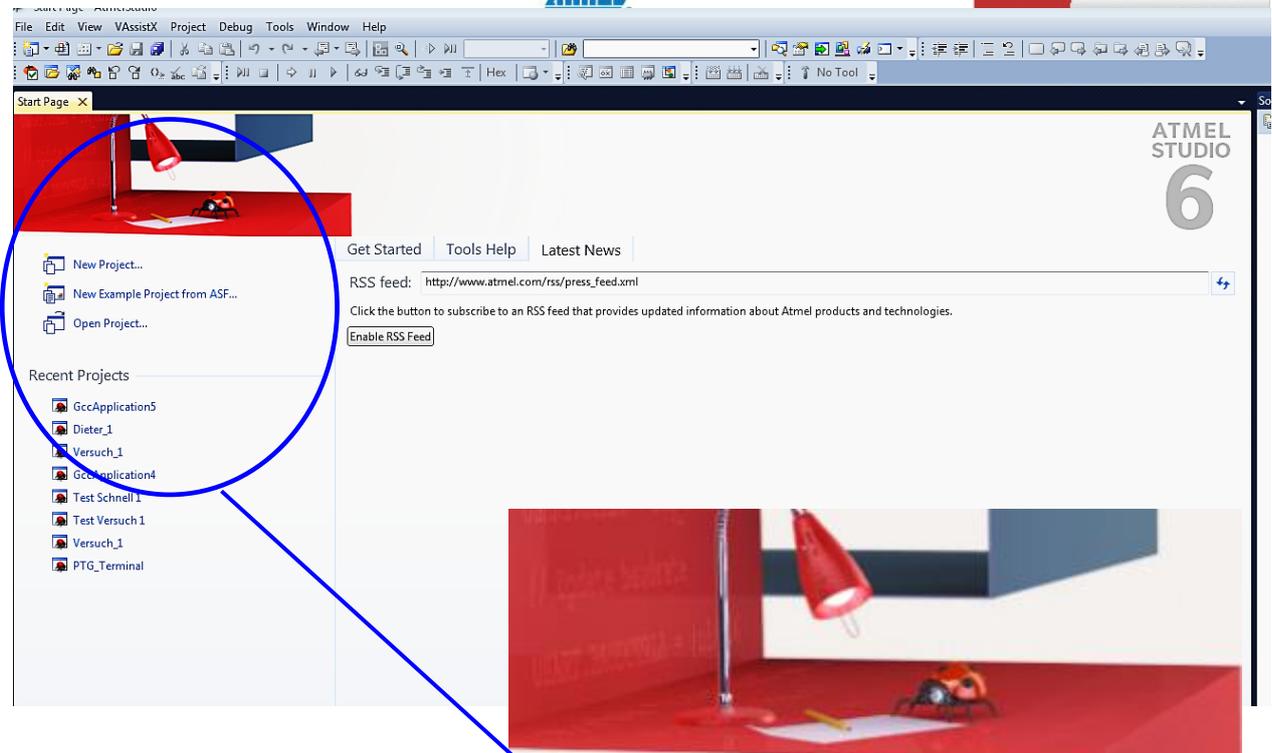
Der Name *Nibo 2* wird mit ausdrücklicher Genehmigung durch die Firma *nicai-systems* verwendet.

1. Atmel AVR Studio 6 - Neues Projekt

Nach dem Start von Atmel AVR Studio 6 erscheint der Startbildschirm

Bitte das Tut vorher durchlesen!
Für die Richtigkeit übernehme ich keine Gewähr.

Danach erscheint die erste Seite vom Studio 6 mit den Auswahlmöglichkeiten

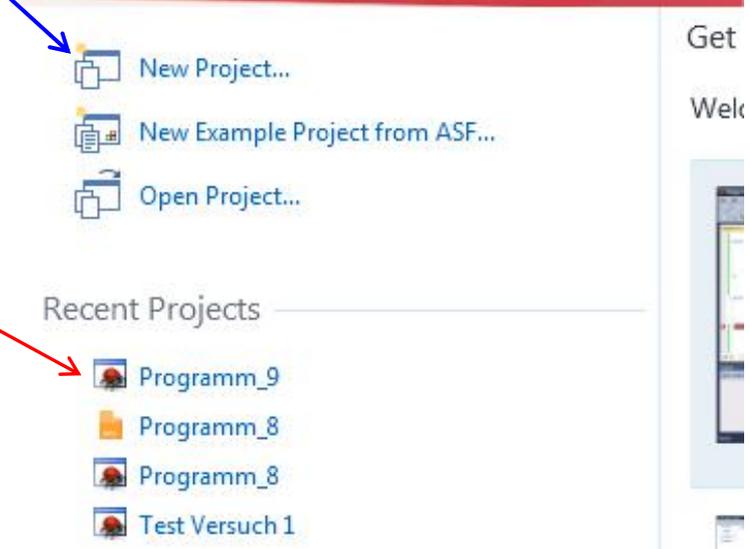


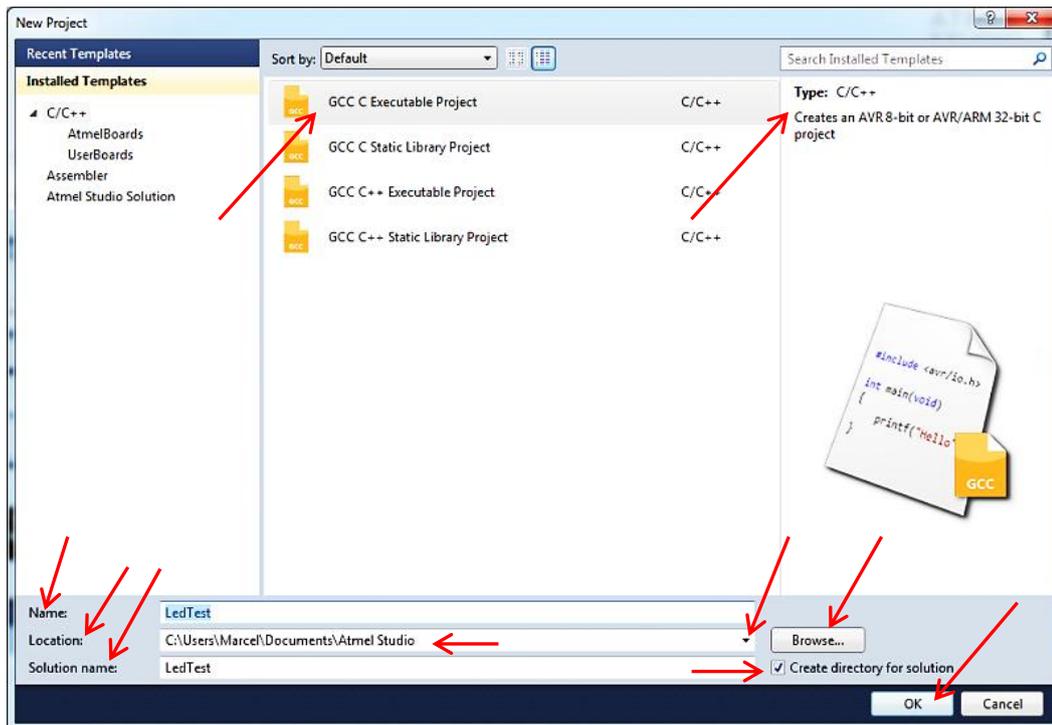
Auswahl eines neuen Projektes
(**New Project**)

Auswahl eines vorhandenen Programmes

1

Nach dem Anklicken von „**New Project**“ erscheint diese Seite. Dort kann man auswählen, was





man genau machen will. Wir wählen das oberste „**GCC Executable Projekt**“. Rechts daneben steht bereits, was man machen kann.

Name - Angabe eines neuen Namens für das Projekt / Programm 2

Location - Angabe eines Speicherortes. Mit der Auswahl (Ende Zeile) kann man zwischen verschiedenen vorhandenen Speicherorten wechseln.

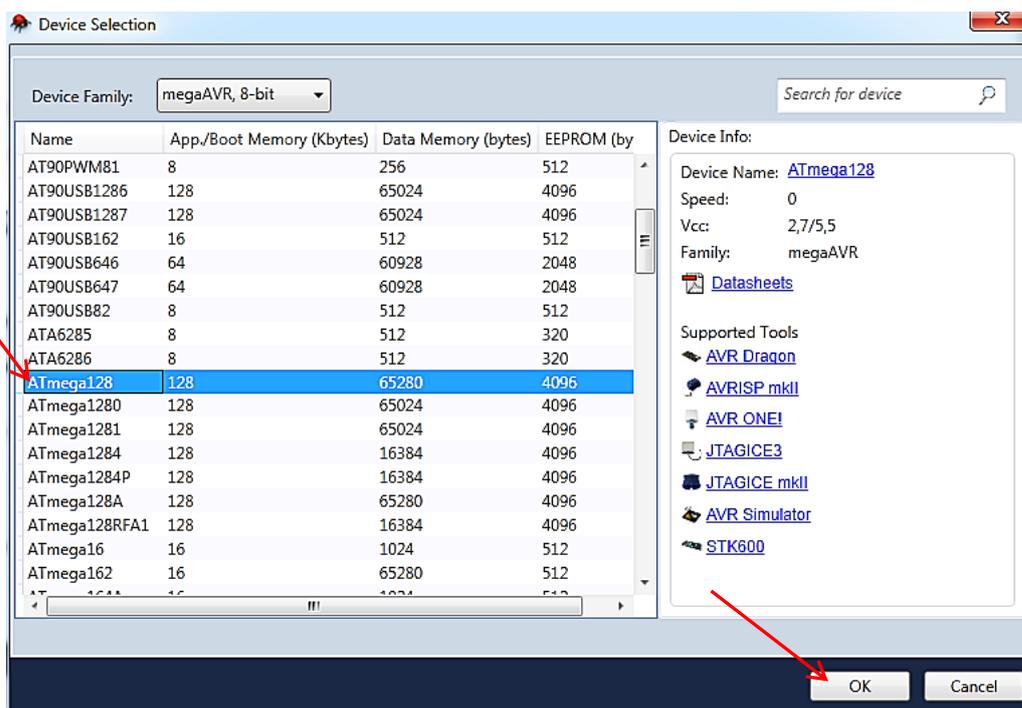
Solution name - erscheint automatisch, wenn Hacken gesetzt (erstellt neues Verzeichnis)

Browse - Auswahl eines neuen Speicherortes (Verzeichnis)

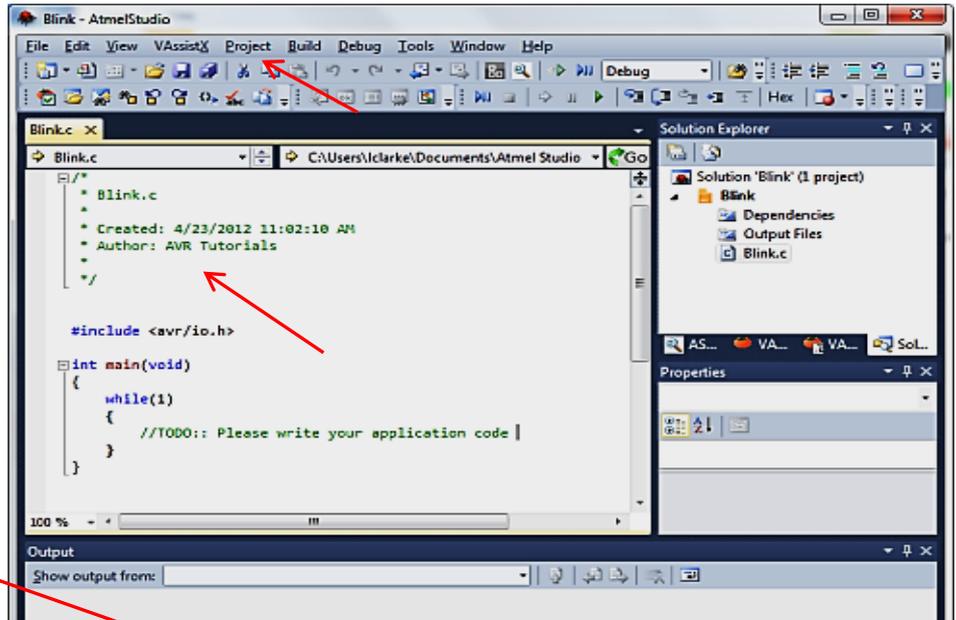
Mit **OK** bestätigen 3

Im nächsten Bild wählt man einen Prozessor aus. z.B. den **ATmega 128**

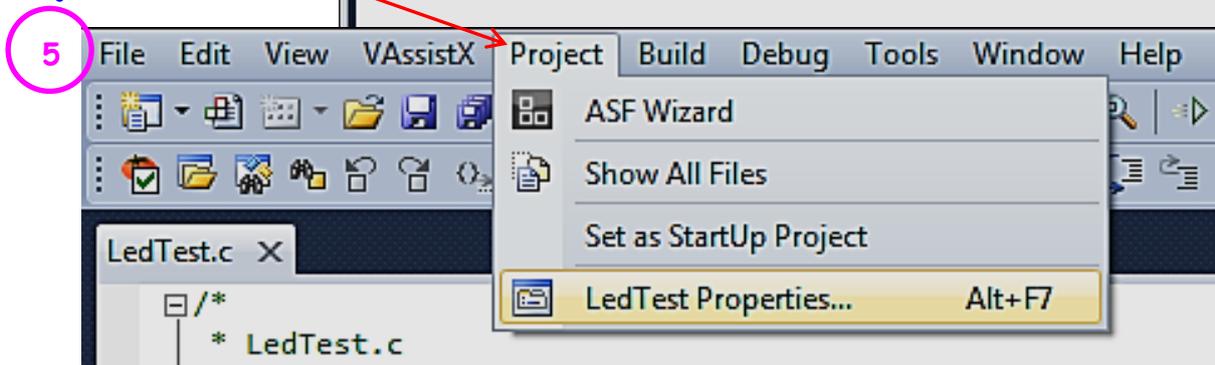
Weiter geht es mit **OK**



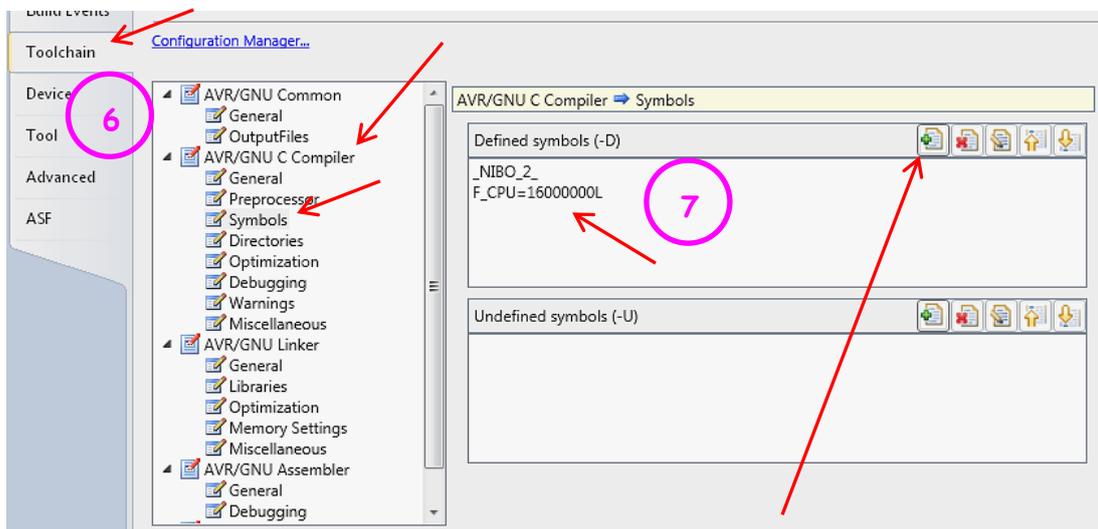
Es erscheint dieses Fenster. Hier kann der C-Code eingegeben werden. Es wird bereits ein kurzer Code vorgegeben.



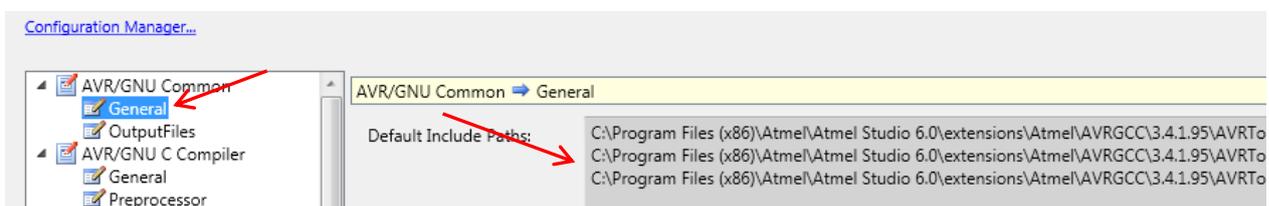
Danach wähle ich den Reiter „Project“ und weiter mein aktuelles Projekt



Danach kann ich „Toolchain“ wählen und meinen „AVR/GNU Compiler“ wählen. Durch Anklicken von „Symbols“ kommt man zu dieser Ansicht.



Danach kann man „Add Item“ anklicken und eine Namen / Wert / Bezeichnung eingeben.



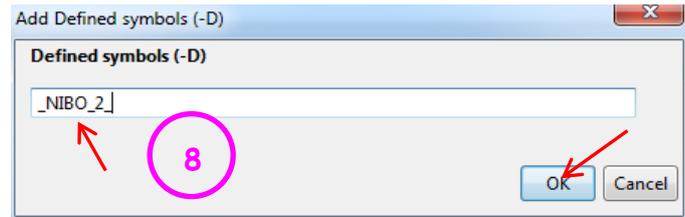
Unter **General** kann ich noch mal kontrollieren, an welcher Stelle im Computer meine **Includes** und **Libraries** stehen. Zunächst müssen wir aber die **Defined symbols** eingeben.

Dazu muss ich das **Add Item** anklicken und dieses Fenster erscheint:

In diesem Fenster bitte „ **_NIBO_2_** „ eingeben und mit **OK** bestätigen.

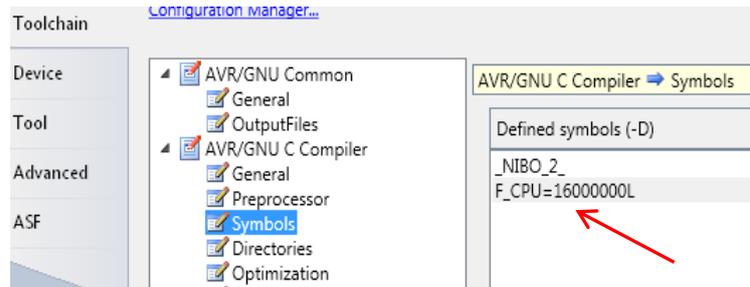
Danach noch mal „ **Add Item** “ anklicken und „ **F_CPU=16000000L** “

eingeben und mit **OK** bestätigen. Wenn ich das NDS3 verwende, muss ich hier noch zusätzlich eingeben: **NIBO_USE_NDS3**. (Bitte Frequenz richtig eingeben: **16 000 000**)

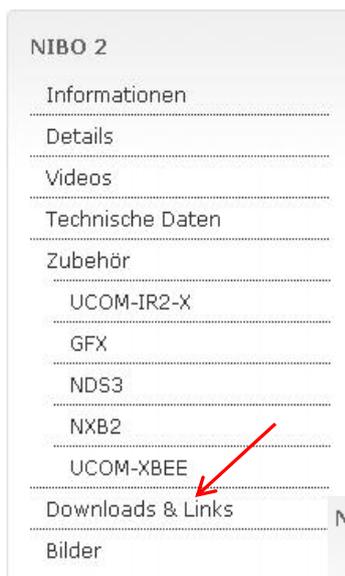


An dieser Stelle erscheint nach OK „ **_NIBO_2_** “ .

Darunter erscheint nach dem zweiten Eintrag „ **F_CPU=16000000L** “ ..



Wir brauchen aber noch unsere Dateien zum Nibo 2. Dazu gehe ich auf die Seite von Nicai (<http://www.nicai-systems.com/de/nibo2.html>) und gehe unter Downloads, weiter zu



Roboter-Bausatz NIBO 2



Nibo Library sourceforge und folge dem Link. Dort kann ich die aktuelle Datei laden.

Zur Zeit ist es die

Nibolib 20110603.msi

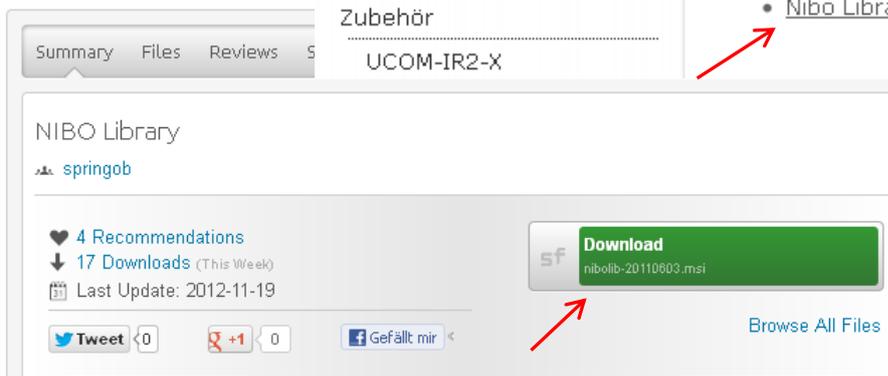
Diese lade ich auf meinen Computer und installiere sie. Da sie die Endung msi hat erfolgt die Installation automatisch oder ich kann wählen.

ROBOTER-BAUSATZ NIBO 2 - DOWN

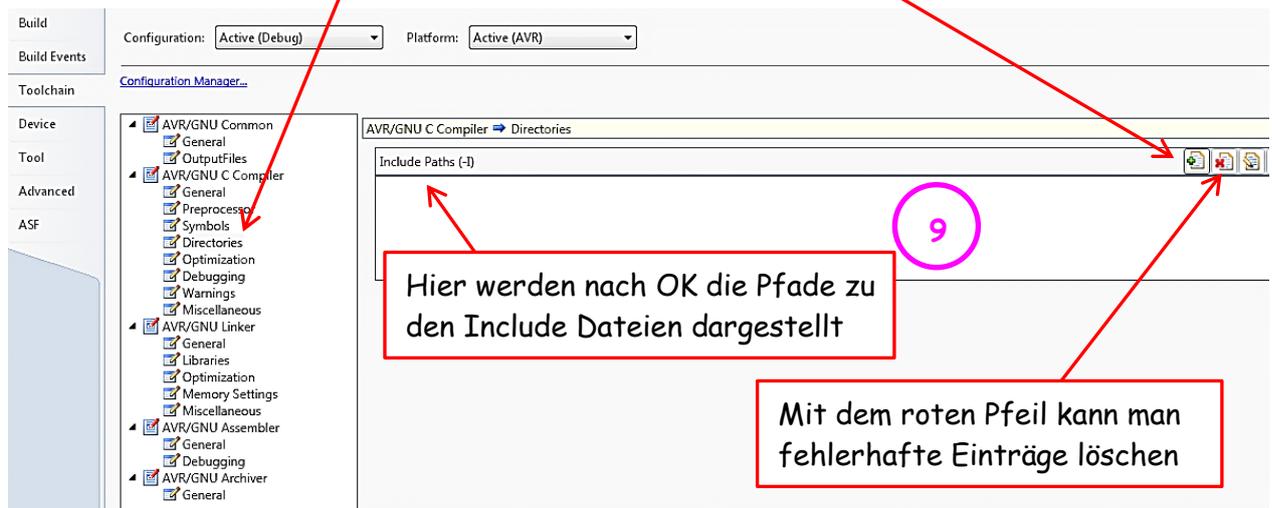
Dokumente & Downloads

- [Bauanleitung Nibo2 \(deutsch\)](#)
- [Bauanleitung Nibo2 \(englisch\)](#)
- [C Programmier tutorial Nibo2](#)
- [Schaltplan zum Nibo2](#)
- [Doku zur Nibo Library](#)
- [Nibo Library **SOURCEFORGE**](#)

Home / Browse / System Administration / Hk

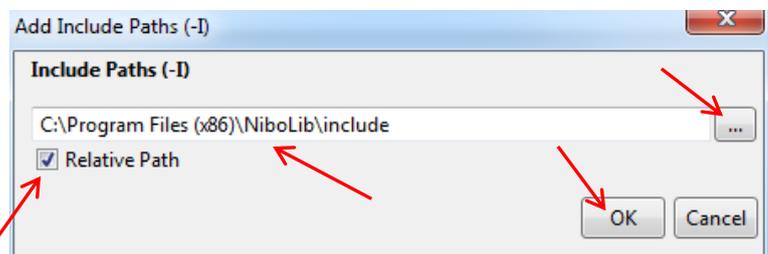


Als nächstes geben wir unsere Pfade zu den notwendigen Dateien an. Wir gehen zurück zum Studio 6 und klicken **Directories** an und weiter mit „**Add Item**“



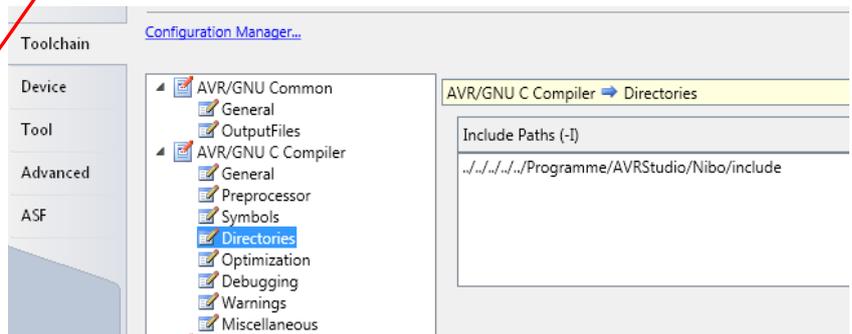
Danach erscheint dieses Bild:

Man kann hier den Pfad eingeben, an dem sich die **Include** Dateien vom Nibo 2 befinden oder man kann sie auch suchen (...) und die Verzeichnisse aufrufen. Weiter mit **OK**.



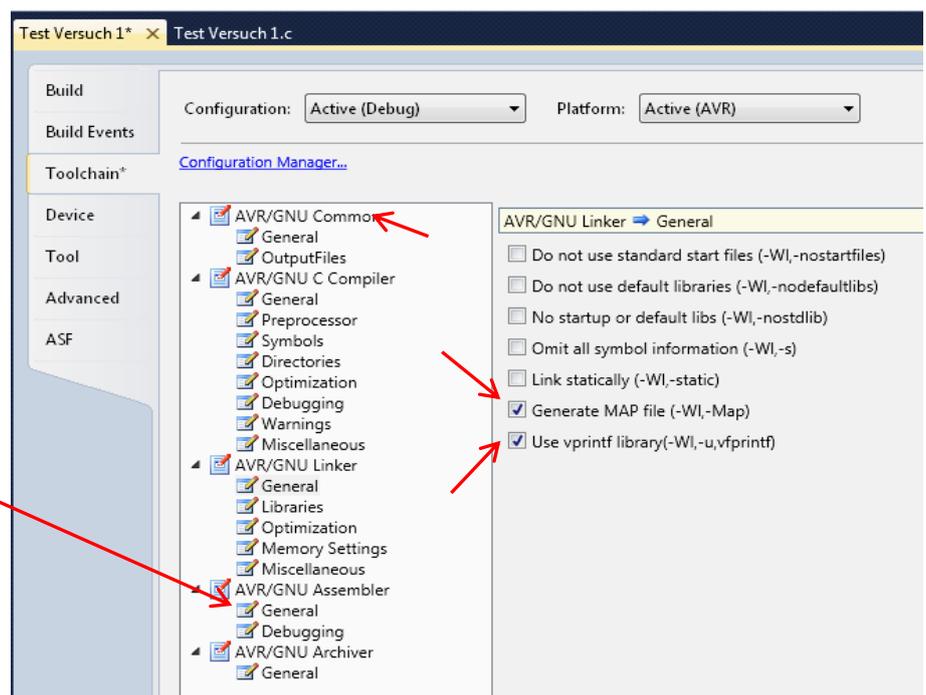
Das Häkchen **Relative Path** bitte unbedingt entfernen.

Bei einem mehrfachen Aufruf des Programmes führte es bei mir zu Programmabstürzen.



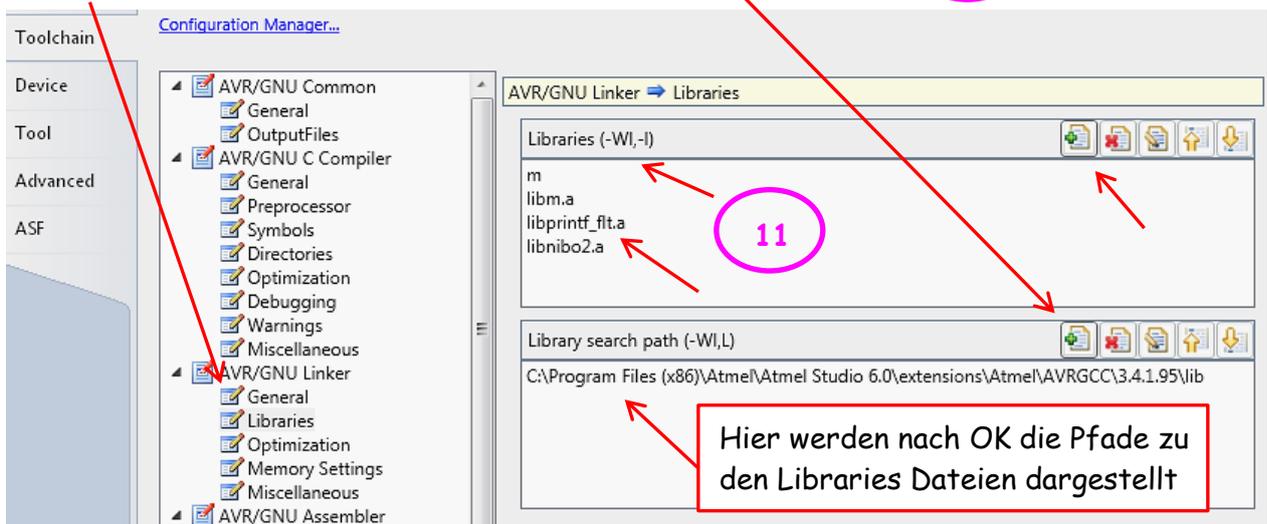
So könnte es danach bei euch aussehen.

Danach bitte unter **AVR/ GNU Linker** „**General**“ anklicken und die Häkchen setzen bzw. kontrollieren.



Danach bitte bei „**Libraries search path**“ „**Add Item**“ anklicken.

10



Danach erscheint dieses Bild:

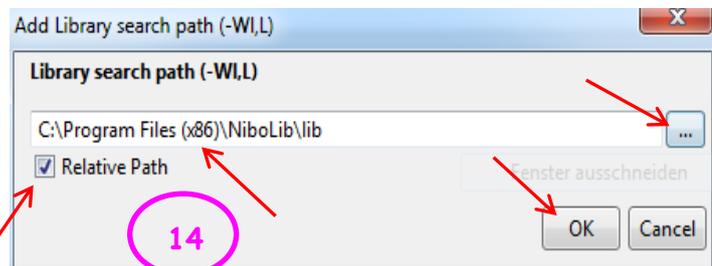
Man kann hier den Pfad angeben, an dem sich die **Library** Dateien vom Nibo 2 befinden sind oder man kann sie auch suchen (...) und die Verzeichnisse aufrufen.

Weiter mit **OK**. (by achim S.)

Den Hacken Relative Path wieder unbedingt löschen !

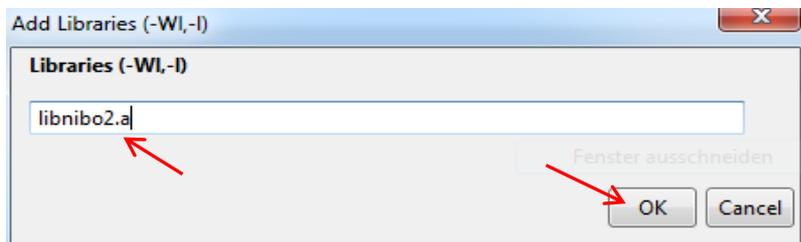
Danach bei **Libraries(-WI,-I)** wieder „ **Add Item** „ anklicken

Im nächsten Bild bitte eingeben und jeweils mit **OK** bestätigen:



- **libm.a**
- **libprintf_flt.a**
- **libnibo2.a**

12



zusätzlich kann noch kommen:
(je nach Programm)

- **libuart-0.a**
- **libnds3.a**

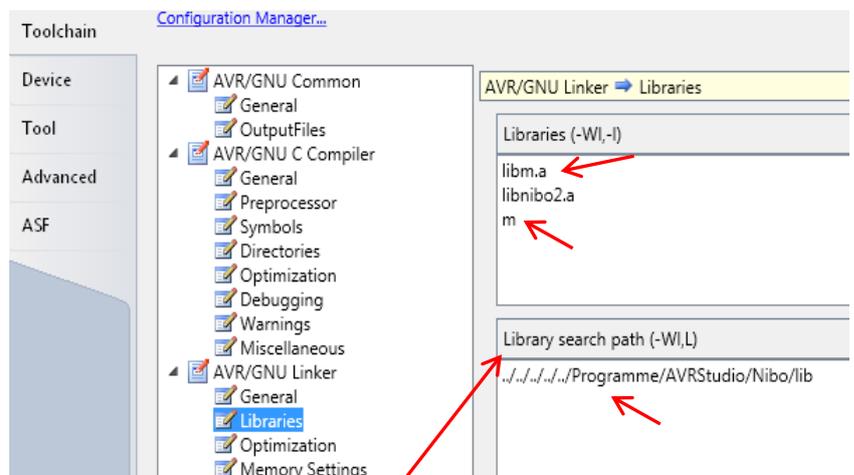
13

Nach dem wir alles eingeben haben, könnte es so aussehen. Das **m** erscheint automatisch. Die Funktion ist mir nicht klar.

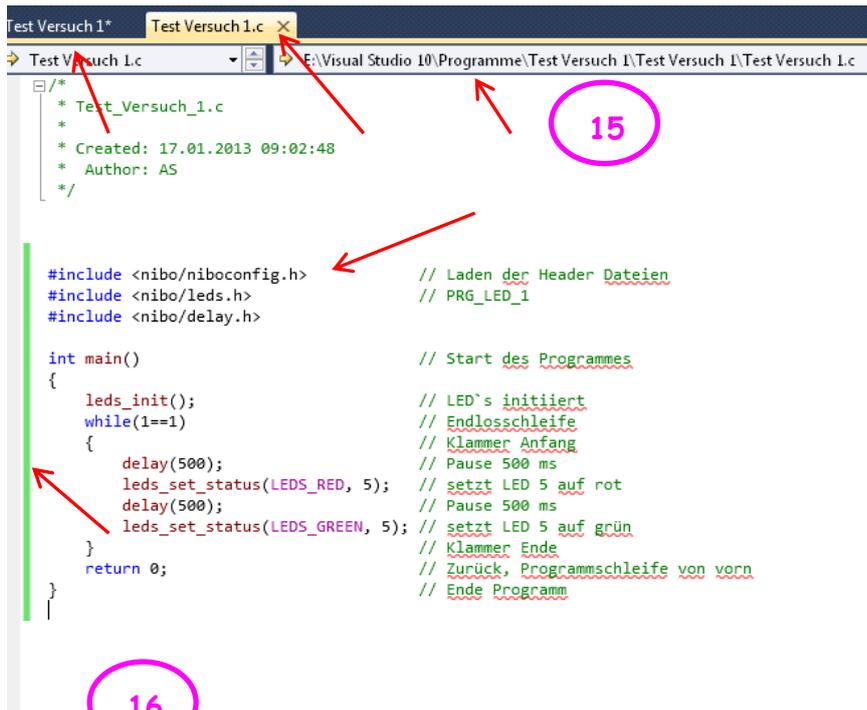
Das Programm **WinAVR** ist nicht mehr notwendig.

Eingab eines neuen Pfades für Libraries und Includes:

- alten Pfad anklicken (wird blau)
- mit dem roten Feld mit (Kreuz) löschen
- und neuen Pfad eingeben

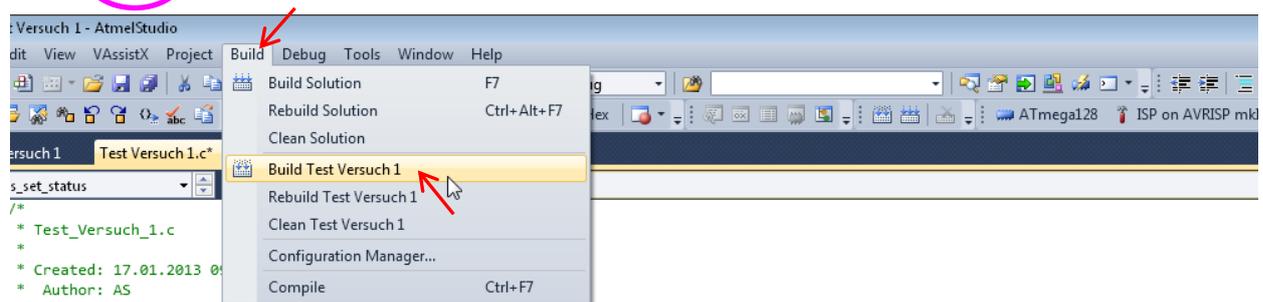


Damit haben wir die wichtigsten Einstellungen vorgenommen.
Gehen wir zu z.B. **Test Versuch1.c**

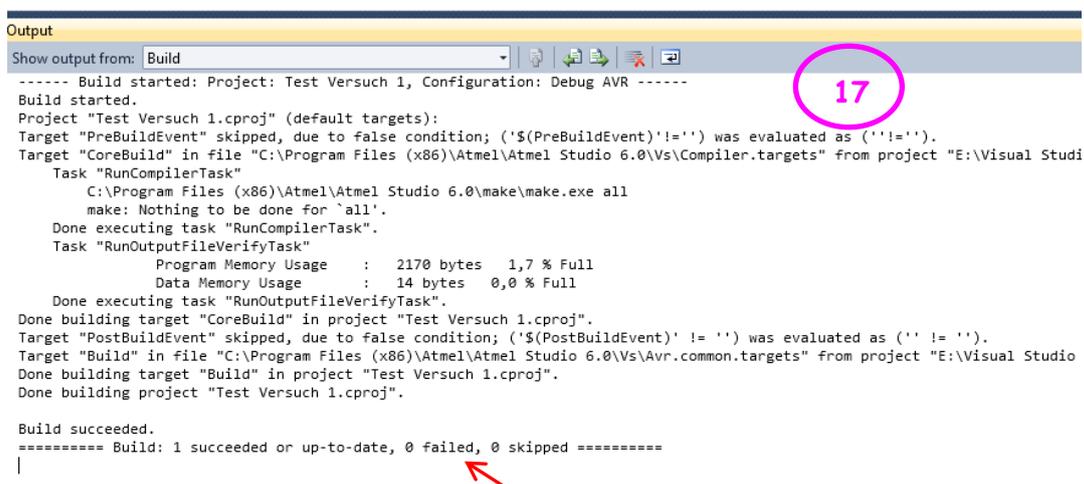


Jetzt kann man ein Programm eingeben. Man kann auch die bereits vorhandene Teile löschen.
An der oberen Kante wird dabei der Name des Projektes und der Speicherort angezeigt.

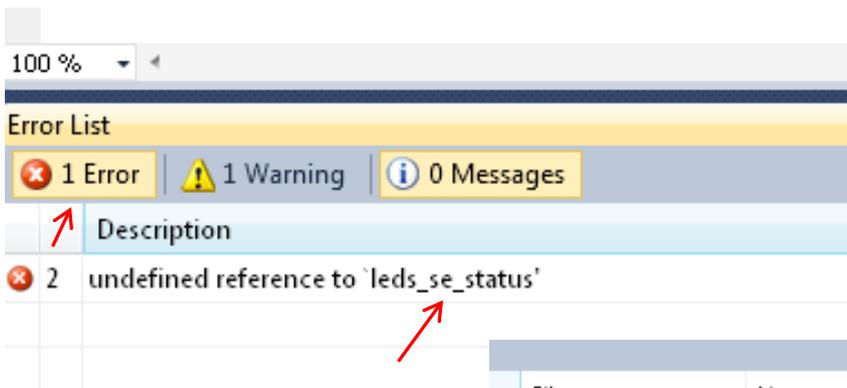
Der Streifen an der linken Seite grün zeigt an, ob das Programm abgespeichert ist oder nicht.
Grün bedeutet abgespeichert. Gelb bedeutet nicht abgespeichert.



Danach kann ich über „ **Build (Test Versuch1)** „ oder dem **Icon** auf der Leiste veranlassen, das das Programm auf Fehler überprüft und übersetzt wird.
Sollte das Programm Fehlerfrei sein, erscheint das Bild mit den entsprechenden Angaben:



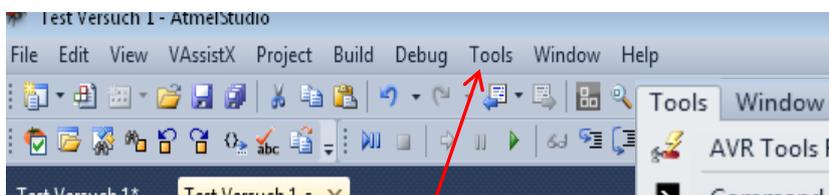
Fehler werden unten angezeigt.



Sollte ein Fehler im Programm sein und eine Übersetzung nicht möglich, so wird ein „**Error**“ angezeigt. Dabei erscheint auch der genaue Fehler. **18**

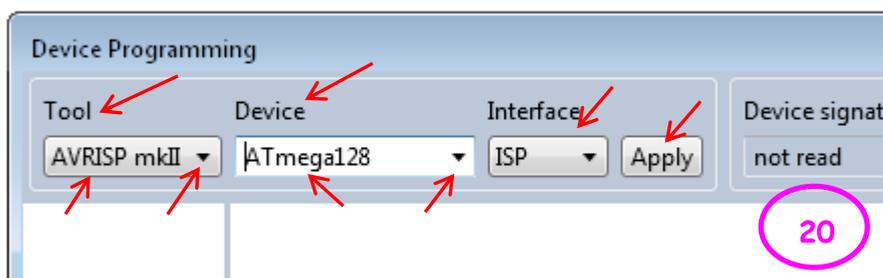
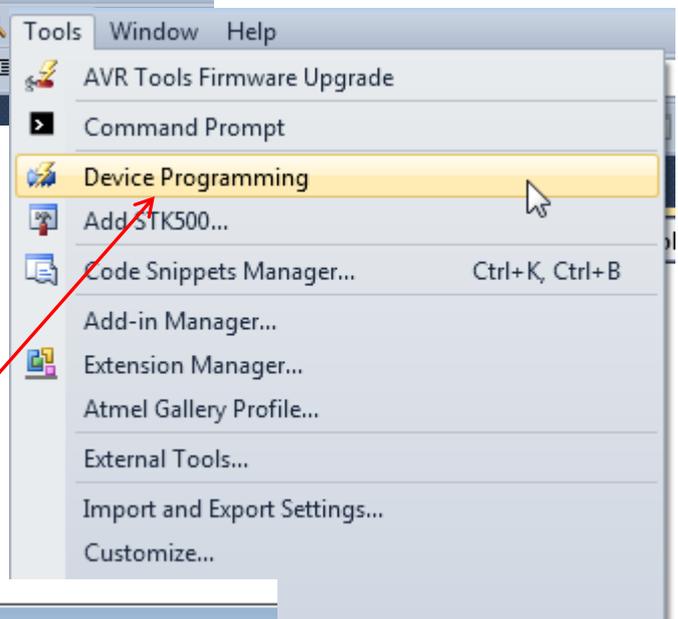
Auf der rechten Seite wird zusätzlich noch die Zeilen Nr. angegeben. (Fehlt ein „†“)

File	Line	Column	Project
Test Versuch 1.c	20	1	Test Versuch 1



Kommen wir zum nächsten wichtigen Punkt. Wir wollen das Programm an den Nibo2 schicken. **19**

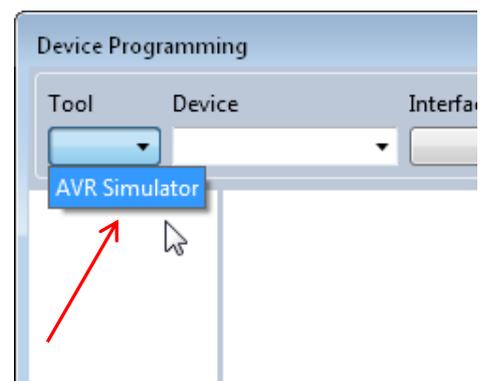
Das kann ich machen indem ich „**Tools**“ anklicke. Danach erscheint das nebenstehende Bild. Hier kann ich „**Device Programming**“ anklicken



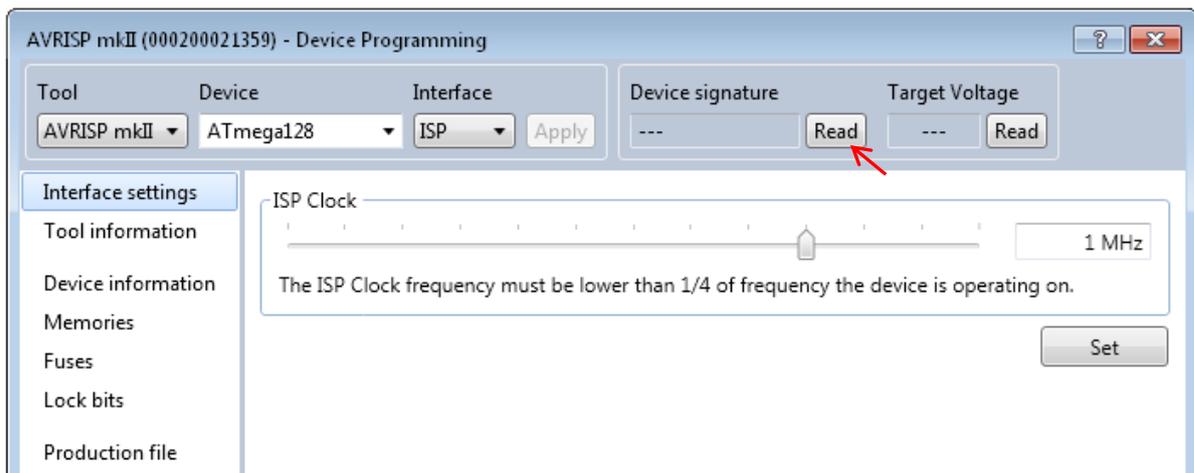
Danach erscheint dieses Fenster. Hier wähle ich unter **Tool** einen entsprechenden Prommer aus. Klicke ich das ent-

sprechende Feld an, so wird mir der angeschlossene Typ mit Seriennummer angezeigt.

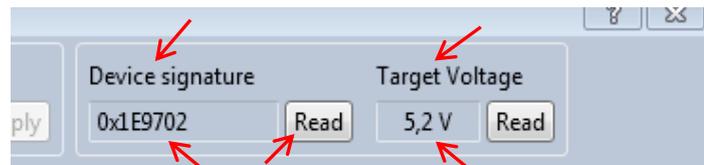
Mit **Device** kann ich meinen Prozessor auswählen. Wenn ich auf die Spitze klicke erscheint eine ganze Liste mit den Prozessoren. Dort wähle ich den **ATmega 128** aus und klicke ihn an. Dieser erscheint dann in meinem Fenster. Als letztes muss ich noch das Interface wählen. Ist kein Prommer angeschlossen, erscheint nur AVR-Simulator. Der **Prommer muss vor dem Start** des Programmes angeschlossen sein!



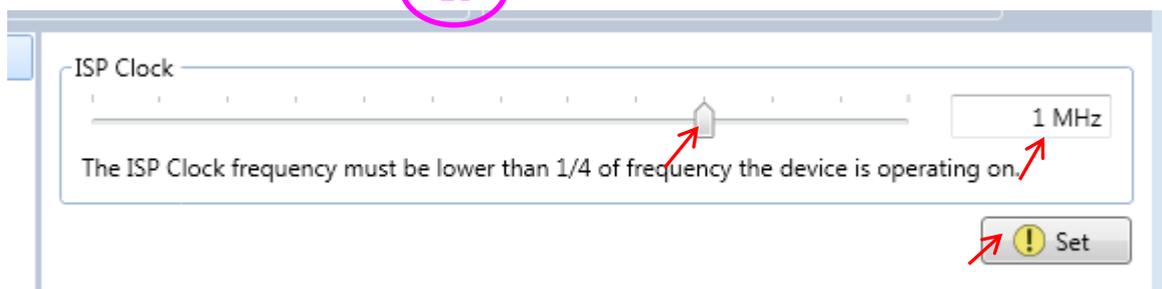
Bei mir erschien sofort **ISP**. Als nächste klicke ich **Apply** an, dann erscheint dieses Fenster.



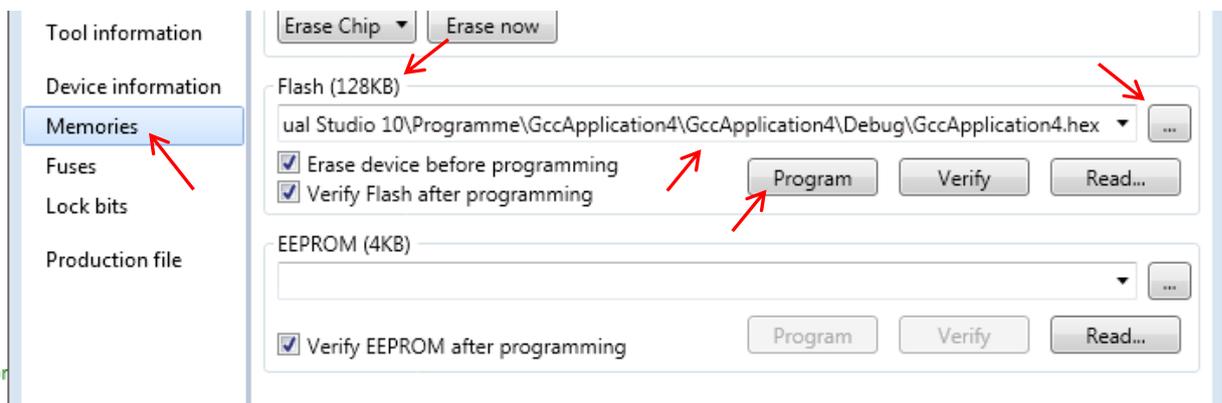
Wenn ich **Read** anklicke erscheint unter **Device signature** die Signatur vom Nibo 2 (**0x1E9702**) und unter **Target Voltage** die aktuelle Betriebsspannung (**5,2V**) des Nibo 2. Damit weiss ich, das eine Verbindung zum Nibo2 besteht. Ist der Nibo 2 nicht in Betrieb, erscheint eine Fehlermeldung.



21



Eine Veränderung der ISP Frequenz ist hier möglich. Dazu kann ich den Regler anklicken und die Maus gedrückt halten und verändern. Die eingestellte Frequenz wird dabei auf der rechten Seite angezeigt. Nach Ende der Einstellung bitte mit **Set** bestätigen. Bitte den Hinweis beachten, das die eingestellte ISP Frequenz maximal **1/4** der Arbeitsfrequenz betragen darf.

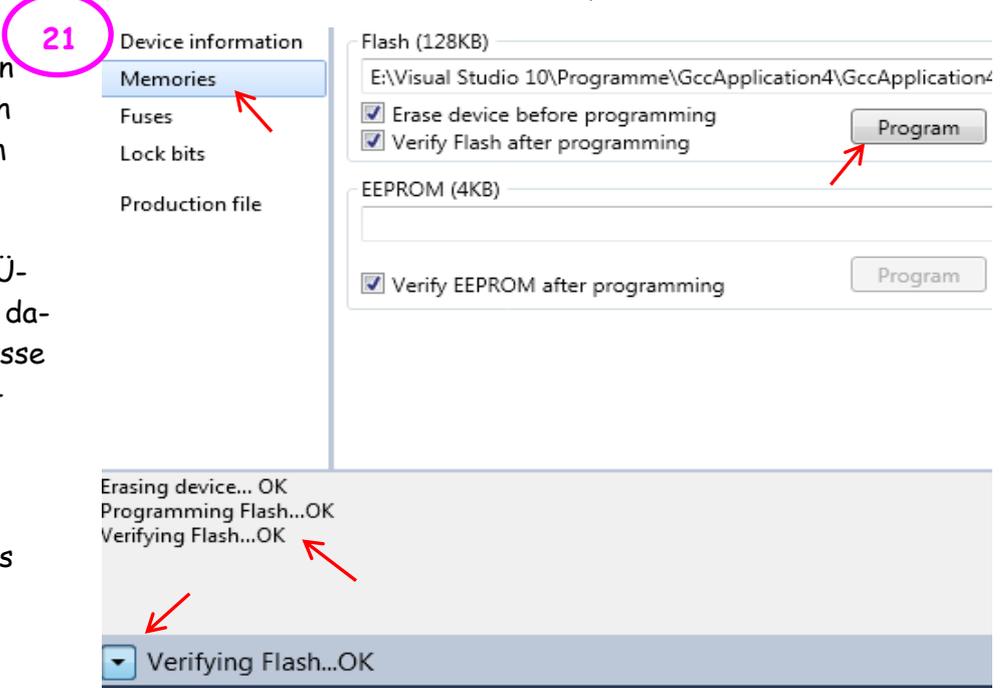


Danach klicke ich **Memories** an und es erscheint diese Ansicht. Unter **Flash (128KB)** erscheint meine Datei. Falls diese nicht kommt oder eine andere, so kann ich diese über einen **Button** suchen.

Durch Anklicken von **Programm** sende ich mein Programm zum Nibo 2.

Eine Kontrolle der Übertragung erfolgt dabei und die Ergebnisse werden unten angezeigt.

Damit erfolgte die Übertragung meines Programmes zum Nibo 2.

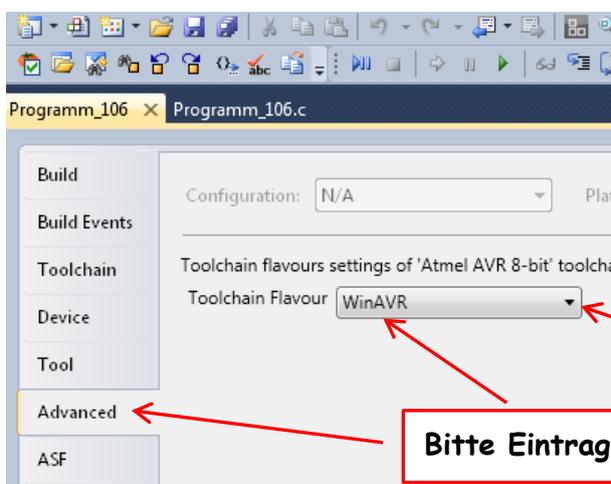
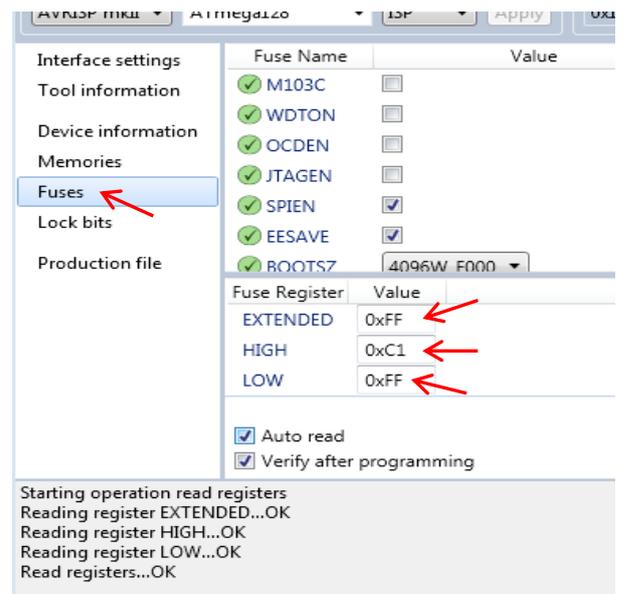


In dem gleichen Feld kann ich auch die Fuse auslesen und verändern. Ich habe sie mal aus meinem Nibo 2 ausgelesen und dargestellt. Das ist die korrekte Einstellung.

Extended 0xFF
High 0xC1
Low 0xFF

Diese Einstellung darf nicht verändert werden !

Sonst besteht Gefahr, das man sich selber aussperrt und eine Nutzung nicht mehr möglich ist. **Achtung - Keine Garantie !**



Bitte kontrolliert unter **Advanced** noch die Einstellung **WinAVR**.

Manchmal kommt es bei mir vor, das der Eintrag **WinAVR** fehlt oder ein anderer Eintrag dort steht. Durch Anklicken der Spitze kann man das richtige Auswählen.

Bitte Eintrag WinAVR kontrollieren

2. Atmel AVR Studio 6

Kurzanleitung - Neues Projekt

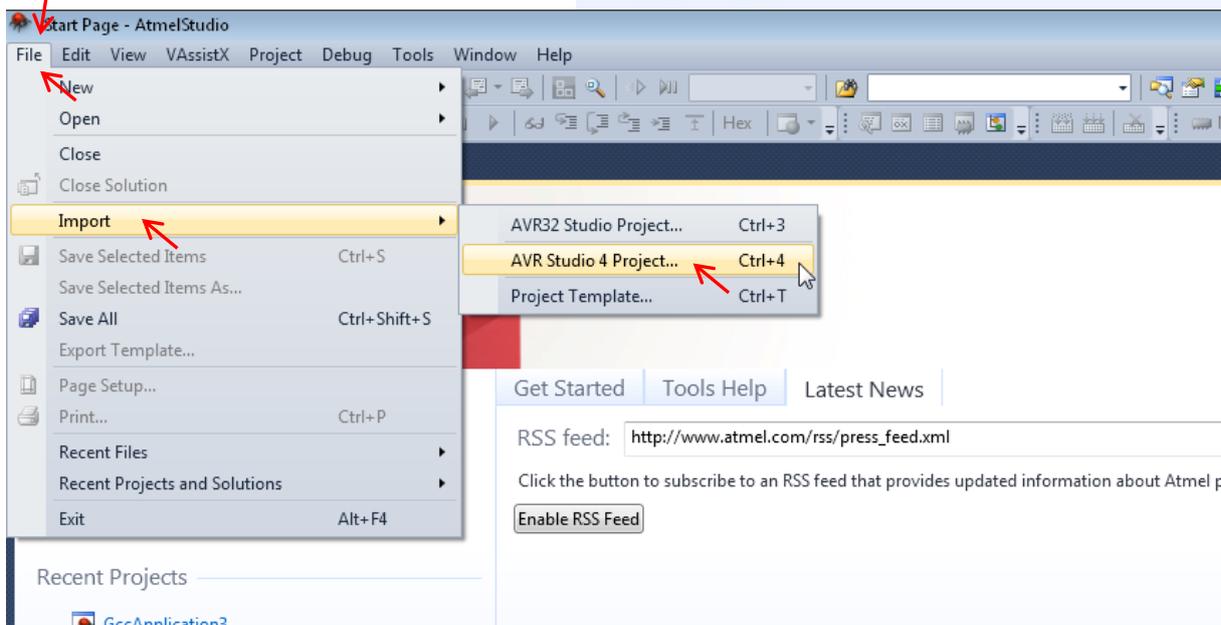
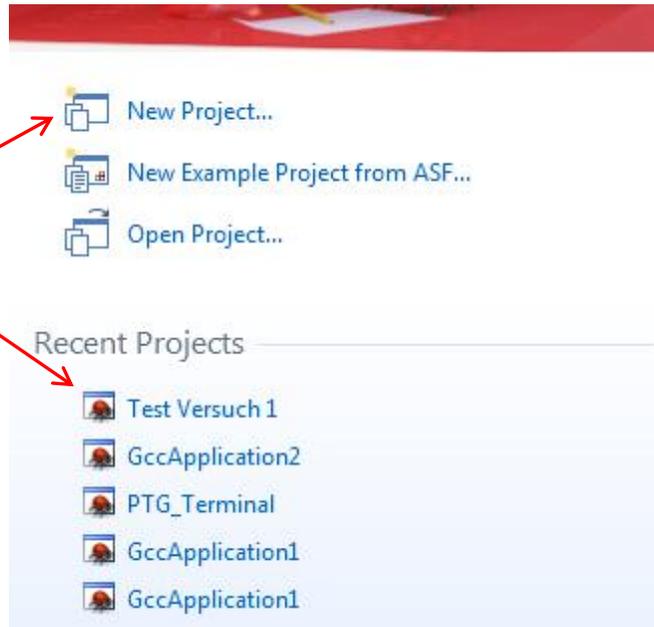
- Atmel AVR Studio 6 **starten** - es erscheint Startbildschirm
Auf dem ersten Startbildschirm - **New Project** auswählen (1)
- Auswahl** **Name** (Eingeben oder Ändern) (2)
Speicherort (Location) (Auswählen oder eingeben) (3)
weiter mit **OK**
- Auswahl in der Leiste **ATmega 128** (4)
weiter mit **OK**
- Auswahl** über Reiter **Projekt** - Auswahl meines **Projektes (Namen)** (Alt+F7) (5)
Auswahl Toolchain (6)
 Auswahl AVR/GNU C Compiler
 Auswahl Symbols
 Defined symbols (-D) - Add Item anklicken (7)
 Eingabe **_NIBO_2_** (8)
 Eingabe **F_CPU=16000000**
 Auswahl Directories
 Include Paths (-I) - Add Item anklicken (9)
 Eingabe des Pfades zu den **Include Dateien** (bei jedem anders)
 Ordner auswählen und mit **OK** bestätigen
 Auswahl AVR/GNU Linker
 Auswahl General (10)
 Häkchen bei **Use vprintf ...** setzen
 Auswahl Libraries (-WI, -l)
 Libraries (-WI, -l) - Add Item anklicken (11)
 Eingabe **libm.a, libnibo2.a, libprint_flt.a** (12)
 Zusätzlich **libnds3.a, libuart-0.a und andere** (13)
 Auswahl Library search path (-WI, L) (14)
 Library search path (-WI, L) - Add Item anklicken
 Eingabe des Pfades zu den **Library Dateien** (bei jedem anders)
 Ordner auswählen und mit **OK** bestätigen
- Eingabe** und **testen** eines Programmes (15)
 Auswahl über den **Pfeil** oder **Seitennamen** und auf Seite wechseln und Code eingeben
 Programm **abspeichern** - Über **Icon (Diskette)** in der oberen Leiste
 abspeichern (16)
 Mit dem Reiter **Build** Programm **Build (Programm)** übersetzen und auf Fehler testen (17)
 Anzeige ohne Fehler oder Fehler suchen (18)
- Übertragung** des Programmes zum Nibo 2
 Auswahl **Tools** und weiter mit **Device Programming** (19)
 Angabe **Prommer** und **Prozessor** (20)
 Kontrolle Signature, Spannung und ISP Frequenz (21)
 Auswahl **Memories** und Kontrolle / Auswahl Datei (22)
 Übertragung zum Nibo 2 (23)
 Kontrolle der Übertragung (24)

3. Atmel AVR Studio 6 Import eines Projekt von AVR Studio 4

Nach dem Start von Atmel AVR Studio 6 erscheint dieses Bild.

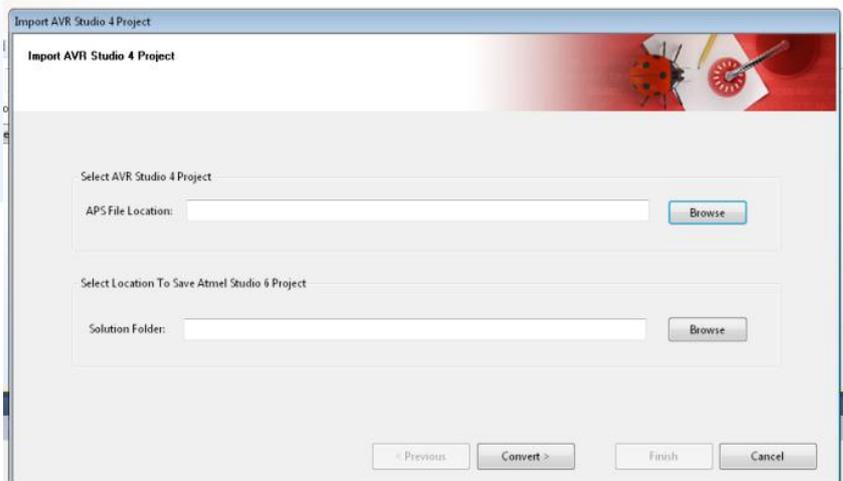
Hier kann ich auswählen, welches **Projekt** ich öffnen will.

Ich kann ein neues Projekt starten oder ein anderes bereits vorhandenes Projekt öffnen oder ein Projekt importieren. Wenn ich ein Projekt importieren will, dann muss ich ...

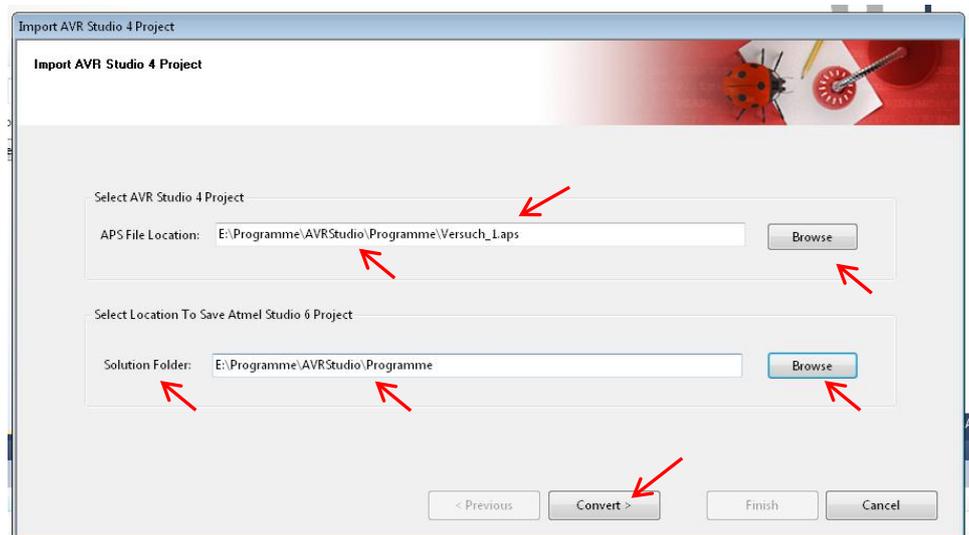


... in der obersten Reihe „File“ wählen, danach „Import“ und „AVR Studio 4 Projekt ..“ anklicken.

Dann kommt dieses Fenster.



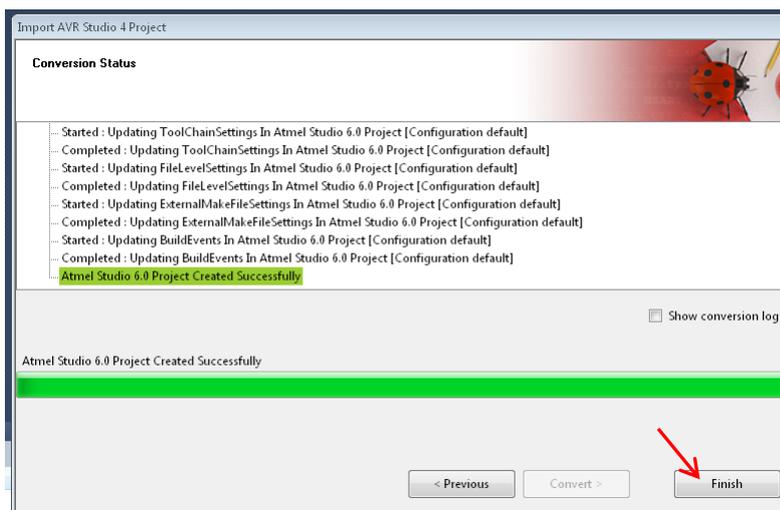
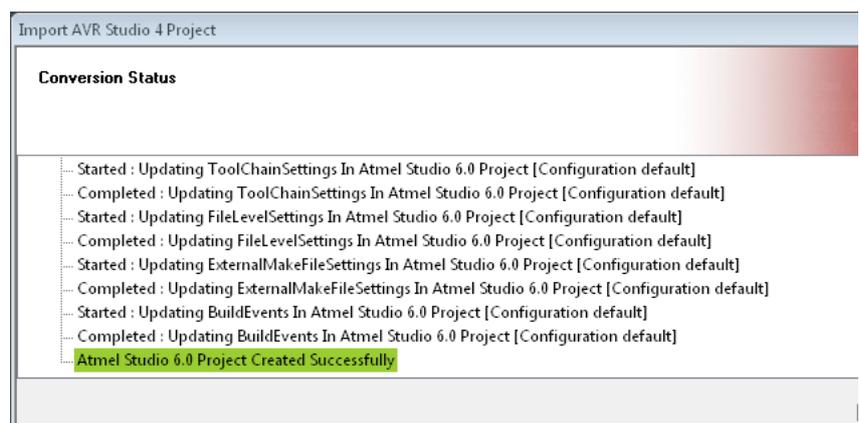
In diesem Fenster kann ich den Namen des Programmes angeben oder über Browse suchen. Dazu muss ich als erstes das Programm suchen bzw. eintragen, das ich von AVR Studio 4 zum AVR Studio 6 übertragen / importieren will.



Dazu wähle ich aus **Select AVR Studios 4 Project** einen Namen aus oder wähle / suche es über **Browse**. Achtet bitte auf die Endung ... **.aps**. Das gewählte Programm / Name und der Pfad muss im Feld **APS File Location** erscheinen.

Mit **Select Location To Save Atmel Studio Project** kann ich das Programm neu abspeichern. Dazu muss der neue Name / Speicherort im Feld **Solution Folder** erscheinen. Ich kann den Namen wieder selbst eingeben oder mit **Browse** suchen und eingeben.

Nach dem ich **Convert** angeklickt habe erscheint eine Sicherheitsabfrage. Diese bestätige ich ebenfalls mit **Yes**



Es erfolgt eine Convertierung des Programmes zum AVR Studio 6. Der Verlauf wird auf dem Display angezeigt.

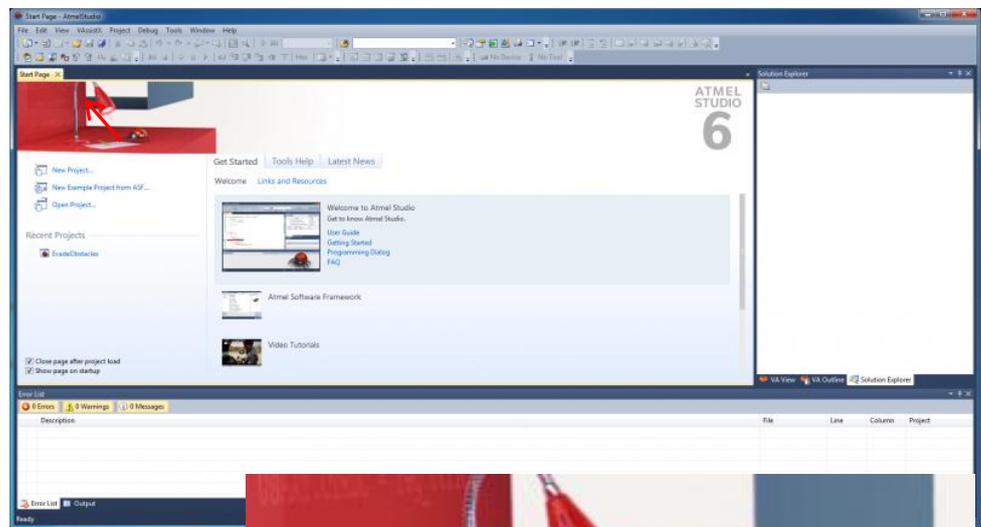
Mit **Finish** kann ich es fertig stellen. Danach geht das Programm in den normalen Ablauf über und ich kann es wie andere auch, weiter einstellen und bearbeiten.

Danach muss ich die Pfade der Include und Libraries Dateien kontrollieren und anpassen. Dazu kann es notwendig sein, die bisherigen Pfade zu löschen und neu einzugeben.

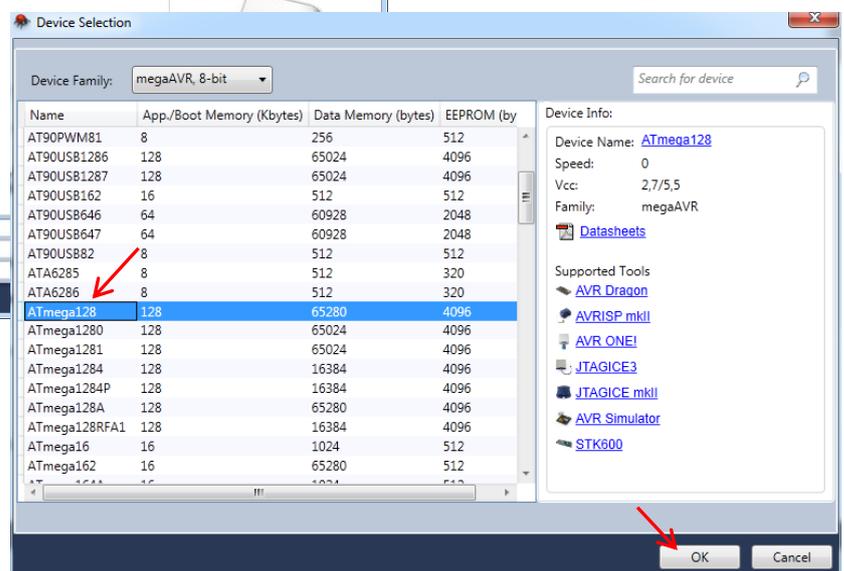
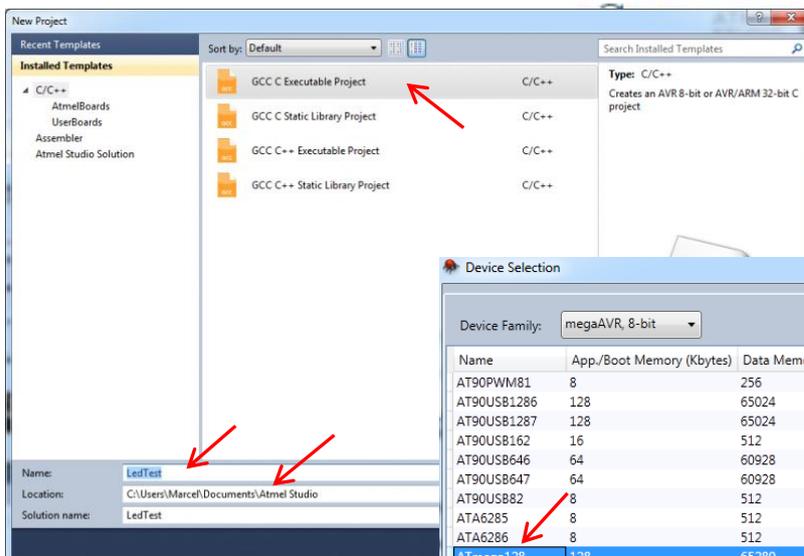
4. Atmel AVR Studio 6 Import eines Projektes als Text

Es ist auch möglich Dateien von anderen Programmen einzufügen. Dazu wird, wie beim normalen Ablauf das Programm gestartet und die notwendigen Einstellungen vorgenommen.

- New Project
- GCC Executable Project
- Name
- Location
- ATmega 128

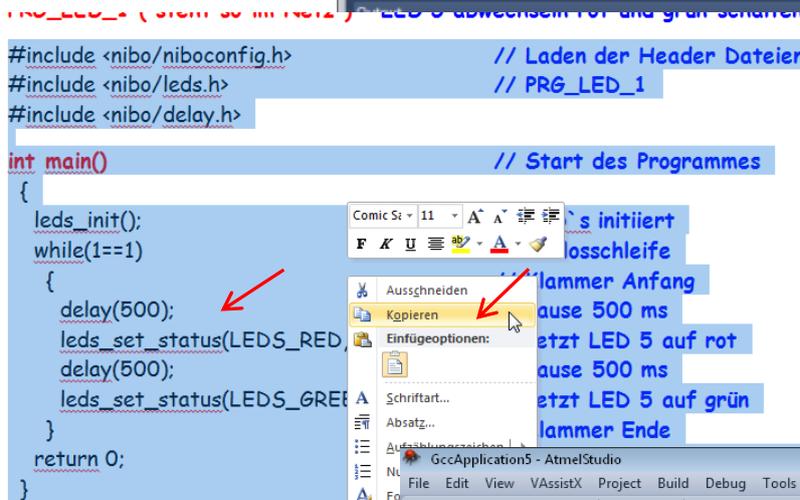
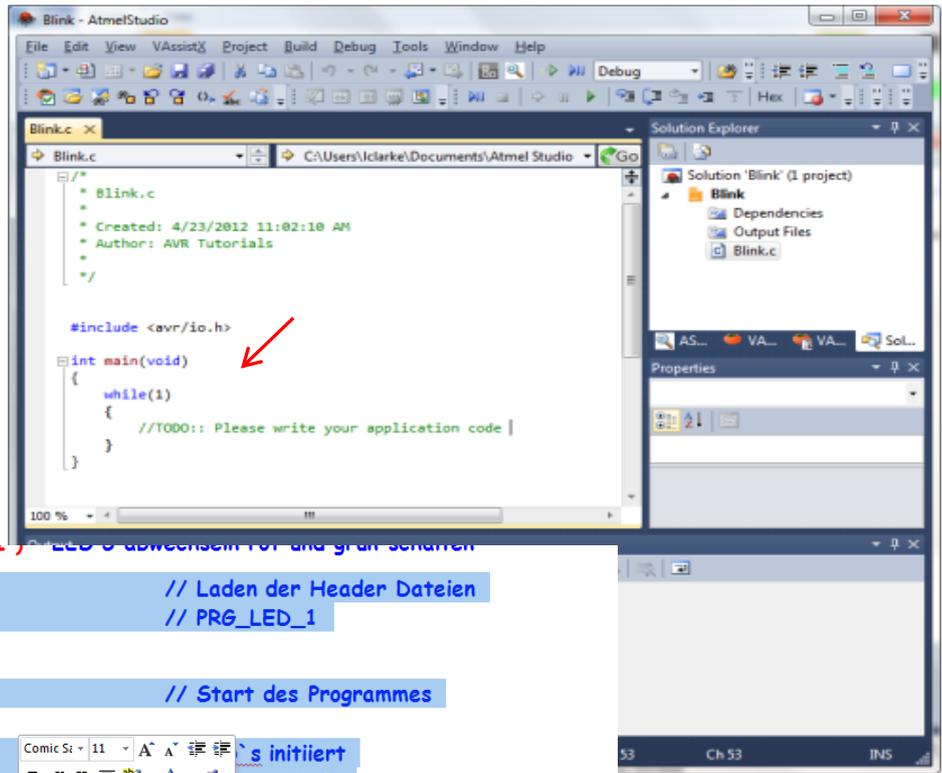


- New Project...
- New Example Project from ASF...



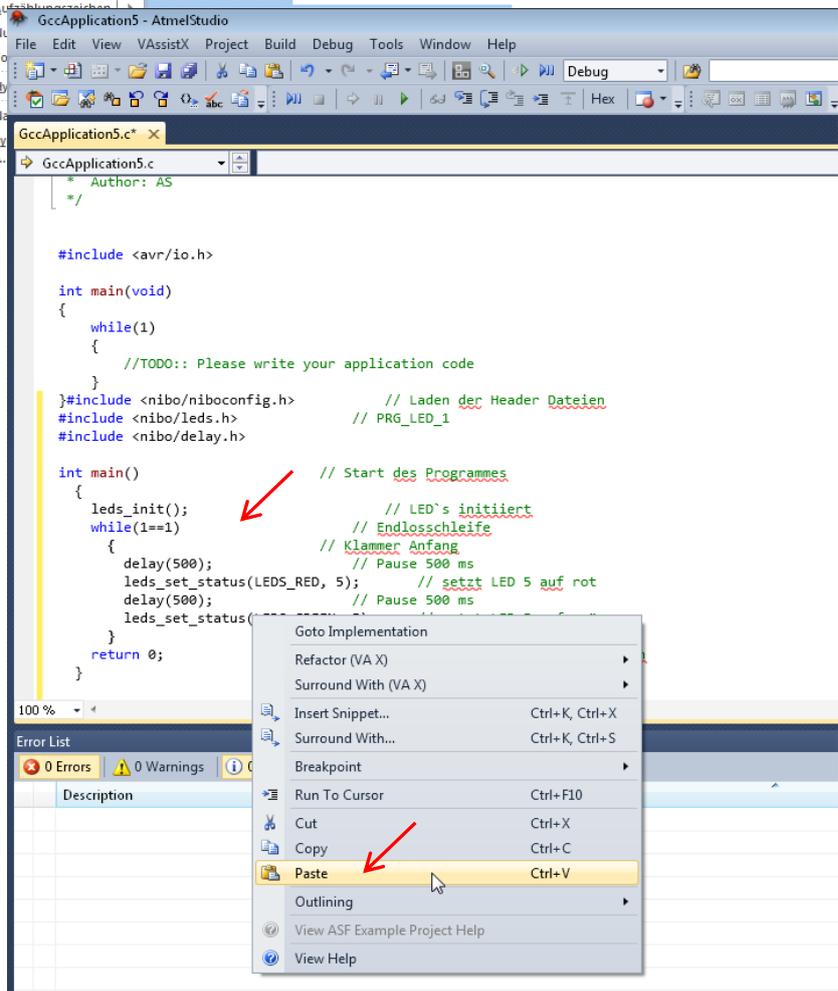
Im nächsten Bild kommt bereits das Programmierfenster. Hier kann man bereits eingaben machen oder ein Programm einfügen.

Dazu kann ich mir z.B. in Word einen Ausschnitt kopieren und in Studio 6 einfügen



In den ersten 3 Zeilen werden die Header Dateien geladen. Das eigentliche Programm beginnt mit der Endlosschleife ab `while(1==1)`.

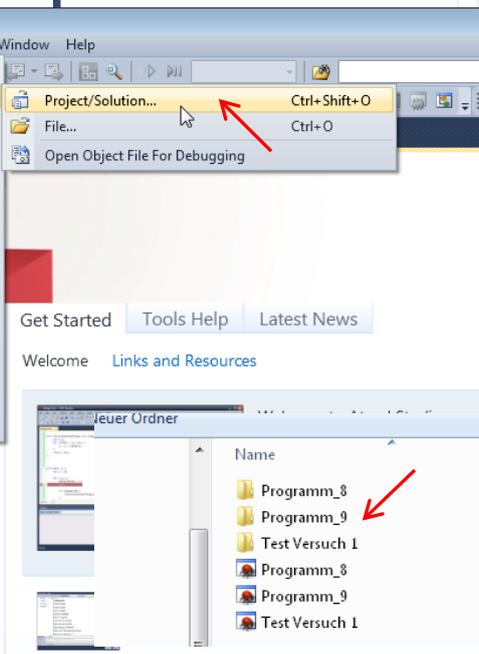
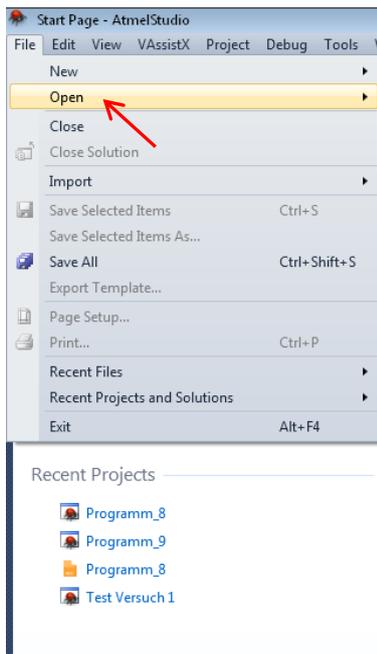
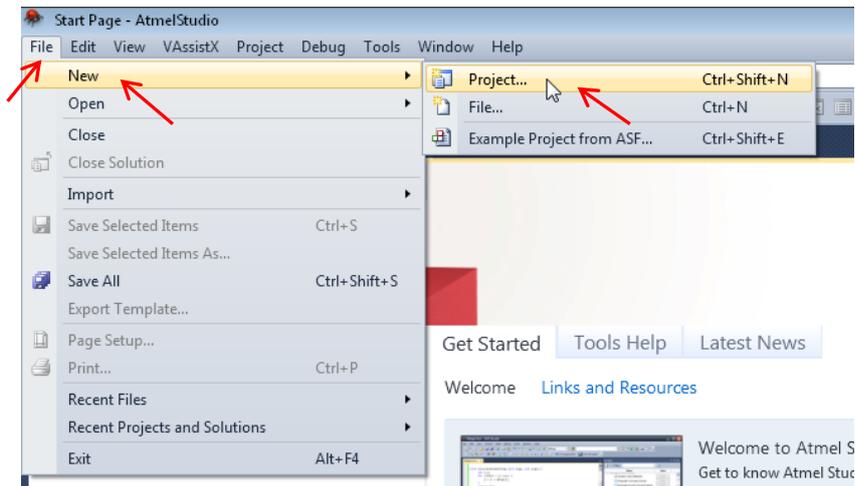
In Word habe ich einen Bereich gekennzeichnet und kopiert (blau hinterlegt). Dann gehe ich zurück zu Studio 4, in mein Programmfenster. Betätige die rechte Maustaste und durch anklicken von Paste wird mein Text eingefügt. Wenn ihr die beiden Teile mit einander vergleicht, so ist es dasselbe. Nicht benötigte Zeilen kann ich löschen. Dadurch kann ich auch Teile von Programmen übernehmen. Der gelbe Streifen an der Seite zeigt mir an, das das Programm noch nicht abgespeichert wurde.



Man kann ein neues Projekt auch durch einen Klick in der **Leiste** starten. Dadurch ist der Ablauf genauso wie oben beschrieben.

Man geht auf **File**, weiter auf **New** und dann auf **Projekt**. Die anderen Auswahlen und Einstellungen sind wie oben.

Weiter ist es möglich ein vorhandenes Programm, das



nicht angezeigt wird, auf dem Rechner zu suchen und zu öffnen. Dazu geh ich auf **Open**, weiter zu **Projekt** und wähle auf meinen Rechner ein **Projekt** aus.

5. Atmel AVR Studio 6 Software und mögliche Fehler

Die notwendige Software ist auf den Seiten von Atmel kostenlos erhältlich. Ein Nachteil dabei ist allerdings, das man seine E-Mail Adresse und „persönliche“ Daten preisgeben muss.

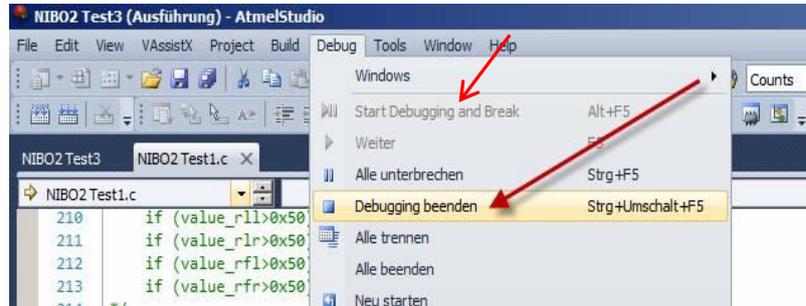
Die angegebene E-Mail Adresse muss korrekt sein, da auf dieser die Adresse mitgeteilt wird, von wo aus das Laden der Software erfolgt.

Bitte nicht an der Grösse erschrecken. Bei mir waren es ca. 780M. Zur Übertragung braucht man dann schon eine schnellere Verbindung. Trotzdem hat es ca. 30 Minuten gedauert.



Das Programm **Win AVR 2010** ... war bei mir nicht nötig.

Der Prommer von Nicali vom Typ **UCOM-IR2** (alte Version) funktionierte bei mir nicht mehr. Ob ein Softwareupdate möglich ist kann ich nicht sagen. Er wurde allerdings vom Gerätemanager erkannt, vom Studio 6 nicht.

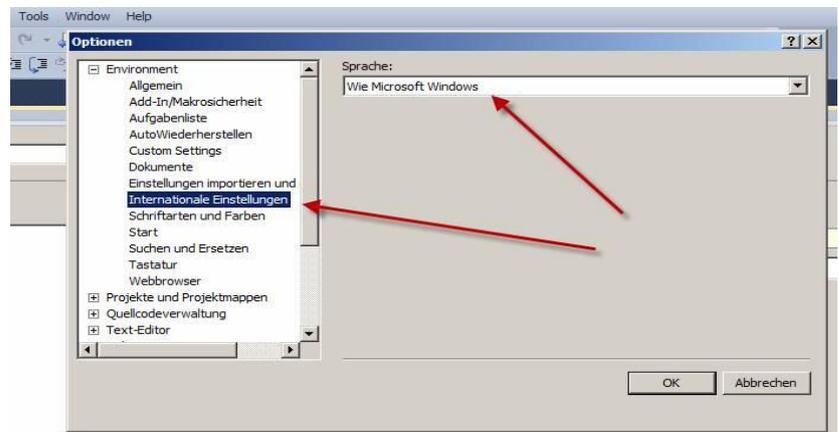


Ein Fehler, den ich auch gemacht habe, ist **Debug**. Hatte diesen zu Anfang angeklickt und mich gewundert, warum alles so lange dauert.

Beim Nibo 2 ist es aber nicht ratsam, da sehr viele **delay** und **Ausgabe**-Anweisungen in den Programmen enthalten sind. Dadurch dauert es länger, bis der Prozessor-Simulator an der gewünschten Stelle angekommen ist. Beenden kann man das mit **Debugging beenden**.

Eine Umstellung der Sprache soll möglich sein. Über **Optionen** und **Internationale Einstellungen** kann man es ändern. Bei mir hat es leider nicht funktioniert.

Ein Teil der Bilder stammt von Jim, besonders die mit den deutschen Bezeichnungen. Danke für seine Hilfe.



6. Atmel AVR Studio 6 - Windows 8

Von Georg habe ich die folgenden Infos erhalten:

Nach der Installation vom ATmel AVR Studio 6 unter Windows 8 treten beim Starten einige Fehlermeldungen auf.

Beseitigung der Fehlermeldung (Achtung Administratorrechte erforderlich):

- Die Programme **atmelstudio.exe** und **avrdbg.exe** mit der rechten Maustaste anklicken
- Eigenschaften öffnen
- Reiter **Kompatibilität** anklicken
- Das **Häkchen** *Programm als Administrator ausführen* setzen
- **Einstellungen** für alle Benutzer ändern

Zu Linux, Vista oder XP kann ich keine Aussagen treffen.

Falls jemand Fehler findet oder der Ablauf anders ist, bitte ich um eine kurze Mitteilung. Dieses Tutorial darf weiter verbreitet werden. Dabei muss es aber komplett sein.

Fragen und Anregungen an h.j.seeger@web.de

Ich wünsche viel Spass beim Umgang mit dem Nibo 2 und ATmel AVR Studio 6

achim S.